Date: 07/02/2025

1. Technical debt management
2. Code optimization, code quality and maintenance
3. CI/CD deployment
4. Data privacy and compliance
5. Methodologies and best practices in Software dev
6. Networking ports and protocols

1. **Technical Debt Management**
    - **Meaning**: The cost of additional work caused by choosing quick, short-term solutions instead of better, long-term approaches in software development.
    - **Key Points**:
        - Accrues when teams prioritize speed over quality.
        - Can be intentional (e.g., for faster releases) or unintentional (e.g., due to lack of knowledge).
        - Needs regular refactoring and code reviews to manage.
    - **Examples**:
        - Hardcoded values instead of configurable settings.
        - Skipping proper testing to meet deadlines.
2. **Code Optimization, Code Quality, and Maintenance**
    - **Meaning**: Improving code efficiency, readability, and maintainability while ensuring long-term stability.
    - **Key Points**:
        - Follow clean code principles (e.g., SOLID, DRY, KISS).
        - Regularly refactor and remove dead code.
        - Automate code reviews and testing.
    - **Examples**:
        - Using efficient algorithms instead of brute force methods.
        - Reducing redundant database queries to improve performance.

3. **CI/CD Deployment**
    - **Meaning**: Continuous Integration (CI) automates testing and merging code, while Continuous Deployment (CD) ensures automated delivery to production.
    - **Key Points**:
        - Uses tools like Jenkins, GitHub Actions, GitLab CI/CD.
        - Automated testing ensures stability before deployment.
        - Helps in faster releases and bug fixes.
    - **Examples**:
        - A pipeline that runs tests automatically when code is pushed.
        - Blue-green deployment to reduce downtime.

4. **Data Privacy and Compliance**
    - **Meaning**: Ensuring that data is handled securely and in compliance with regulations.
    - **Key Points**:
        - Regulations include GDPR (EU), CCPA (California), HIPAA (healthcare).
        - Data encryption, access control, and anonymization are crucial.
        - Audits and compliance checks are necessary.
    - **Examples**:
        - Encrypting sensitive user data in a database.
        - Implementing role-based access control (RBAC).

5. **Methodologies and Best Practices in Software Development**
    - **Meaning**: Approaches used to structure and manage software development efficiently.
    - **Key Points**:
        - Agile (Scrum, Kanban) focuses on iterative development.
        - DevOps integrates development and operations for efficiency.
        - TDD (Test-Driven Development) ensures reliability.

- **Examples**:
    - Using Agile sprints to deliver features incrementally.
    - Writing unit tests before implementing a function.

6. **Networking Ports and Protocols**
    - **Meaning**: Ports are logical endpoints for communication, while protocols define rules for data exchange.
    - **Key Points**:
        - Ports range from 0-65535; well-known ports (0-1023) are for standard services.
        - Common protocols:
            - HTTP (80) / HTTPS (443) – Web communication.
            - FTP (21) – File transfer.
            - SSH (22) – Secure remote access.
            - SMTP (25) – Email sending.
            - DNS (53) – Domain name resolution.
    - **Examples**:
        - A web server listens on port 80 (HTTP) or 443 (HTTPS).
        - SSHing into a remote server using port 22.